

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-339072

(43) 公開日 平成11年(1999)12月10日

(51) Int.Cl.⁶

識別記号

F I

G 0 6 T 17/00

G 0 6 F 15/62

3 5 0 A

15/00

15/72

4 5 0 A

審査請求 未請求 請求項の数21 O L 外国語出願 (全 44 頁)

(21) 出願番号 特願平11-127506

(22) 出願日 平成11年(1999) 3月30日

(31) 優先権主張番号 09/053591

(32) 優先日 1998年4月1日

(33) 優先権主張国 米国 (U S)

(71) 出願人 599002892

リアル スリー・ディー

アメリカ合衆国、フロリダ州 32825、オ
ーランド、レイク アンダーヒル ロード

12506

(72) 発明者 トーマス エー. ピアッツァ

アメリカ合衆国、カルフォルニア州

95746、グラニット ベイ、チェルシャー

ロード 5046

(74) 代理人 弁理士 平田 忠雄

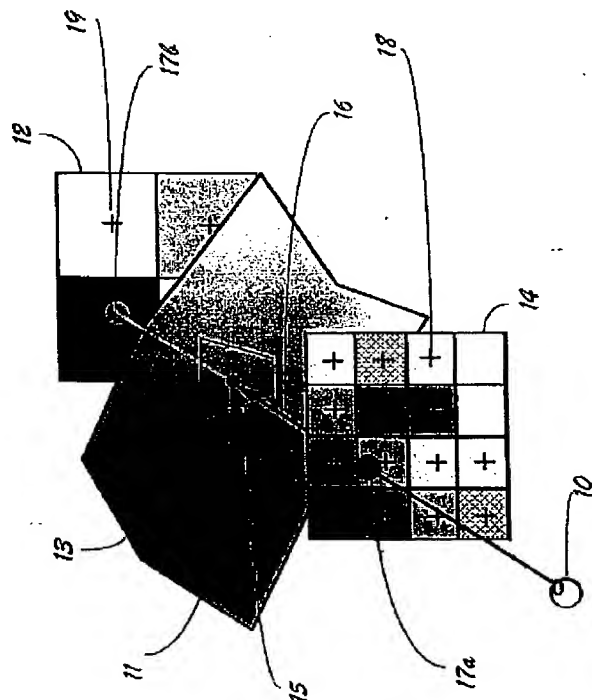
最終頁に続く

(54) 【発明の名称】 テクスチャレベルの詳細ディザリング用方法及び装置

(57) 【要約】

【課題】 ポリゴンを介したテクスチャLOD遷移の可視効果を最小限とする、コンピュータによる効率的な計算方法を提供すること。

【解決手段】 図形要素によりカバーされる各ピクセルについて計算されるLOD値にディザリングオフセット値を加算してディザードピクセルLOD値を得ることにより、可視効果を、最小限に抑えることができる。ディザリングオフセットは、ピクセルスパン内のピクセルのロケーションに基づくテーブルルックアップから生成できる。ディザードピクセルLOD値を、テクスチャードピクセル値を取り出す単一LODテクスチャマップの選択のインデックスとして使用する。テーブルルックアップにおける値を変調することにより、ディザリングオフセット値の範囲を調整できる。



【特許請求の範囲】

【請求項1】オブジェクトのイメージを形成するピクセル値を制御することによりオブジェクトのイメージを生成するコンピュータイメージ生成システムにおいて、各ピクセルの値がテクスチャをオブジェクトに適用するためのテクスチャデータを含む保存データから得られ、前記テクスチャデータがテクスチャデータの複数レベルの詳細（LOD）バージョンからなる一連のテクスチャマップに保存されており、テクスチャを前記オブジェクトに適用するためのLOD間遷移する方法であって、前記オブジェクトのイメージを表す各ピクセル用テクスチャLOD値を計算する工程と、各ピクセル用ディザリングオフセット値を準備する工程と、前記ディザリングオフセット値を各ピクセル用の各計算LOD値に適用して、各前記ピクセル用ディザードピクセルLOD値を得る工程と、各ピクセル用ディザードピクセルLOD値を出力して関連テクスチャLODマップを選択する工程と、を含んでなることを特徴とする、方法。

【請求項2】前記ディザリングオフセット値を準備する工程が、ピクセルのロケーションに基づいて前記範囲のディザリングオフセット値からディザリングオフセット値を選択することを含む、請求項1に記載の方法。

【請求項3】前記選択工程が、選択をピクセルスパンにおけるピクセルのロケーションに基づいておこなうことを含む、請求項2に記載の方法。

【請求項4】前記準備されたディザリングオフセット値を変調する工程をさらに含む、請求項1に記載の方法。

【請求項5】前記変調工程が、前記準備されたディザリングオフセット値を減衰することを含む、請求項4に記載の方法。

【請求項6】前記減衰工程が、制御状態変数に基づくディザリングオフセット値を表すビットをシフトすることを含む、請求項5に記載の方法。

【請求項7】前記減衰工程が、全減衰～ゼロ減衰の別個の工程で設けられる、請求項4に記載の方法。

【請求項8】前記適用工程が、バイアス値を前記ディザードピクセルLOD値に加えることを含む、請求項1に記載の方法。

【請求項9】前記適用工程が、前記ディザリングオフセット値を前記ピクセルLOD計算値に加えることを含む、請求項1に記載の方法。

【請求項10】整数と、切り捨てと丸めのうちの一つによる端数とを含んでなる前記ディザードピクセルLOD値からの整数値を計算することをさらに含む、請求項1に記載の方法。

【請求項11】オブジェクトのイメージを形成するピクセル値を制御することによりオブジェクトのイメージを生成するコンピュータイメージ生成システムにおける、

各ピクセルの値がテクスチャをオブジェクトに適用するためのテクスチャデータを含む保存データから得られ、前記テクスチャデータが複数レベルの詳細（LOD）バージョンからなる一連のテクスチャマップに保存されており、テクスチャを前記オブジェクトに適用するためのLOD間遷移する方法に用いる、テクスチャを前記オブジェクトに適用するときのLOD間遷移をおこなうときのアーチファクトを減少させる各ピクセルに使用するLODを決定する装置において、

テクスチャリングすべきオブジェクトのイメージを表す各ピクセルのディザリングオフセット値を生成するディザパターンファンクション生成器と、ピクセルLOD計算値及び各前記ピクセル用ディザリングオフセット値を受ける結合器であって、前記ディザリングオフセット値を前記ディザパターンファンクション生成器から受け取り、前記ピクセルLOD計算値とディザリングオフセット値とを結合させてディザードピクセルLOD値を生成する、結合器と、を含んでなる装置。

【請求項12】前記ディザパターンファンクション生成器が、テーブルルックアップを含む、請求項11に記載の装置。

【請求項13】前記ディザパターンファンクション生成器が、前記ピクセルのロケーションに基づく前記テーブルルックアップからディザリングオフセット値を選択する手段を含む、請求項11に記載の装置。

【請求項14】前記テーブルルックアップにおけるディザリングオフセット値数がピクセルスパンにおけるピクセル数に等しく、前記選択手段が前記選択を前記ピクセルスパンにおけるピクセルのロケーションに基づいておこなう手段を含む、請求項13に記載の装置。

【請求項15】前記ディザパターンファンクション生成器と前記結合器との間に結合して変調器をさらに含んでなる、請求項11に記載の装置。

【請求項16】前記変調器が、前記ディザリングオフセット値を減衰する手段を含んでなる、請求項15に記載の装置。

【請求項17】前記減衰手段が、制御状態変数に基づいたディザリングオフセット値を表すビットをシフトする手段を含む、請求項16に記載の装置。

【請求項18】前記減衰手段が、全減衰～ゼロ減衰の別個の工程において減衰をおこなう手段を含む、請求項17に記載の装置。

【請求項19】前記結合器が、バイアス値をディザードピクセルLOD値に加える手段を含む、請求項11に記載の装置。

【請求項20】前記結合器が、加算器である、請求項11に記載の装置。

【請求項21】整数と、切り捨てと丸めのうちの一つによる端数とを含んでなる前記ディザードピクセルLOD値からの整数値を計算する手段をさらに含んでなる、請

求項11に記載の装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一般的にリアルタイムコンピュータイメージ生成システムに関する。より詳細には、本発明は、テクスチャマップフィルタリング用手段及び方法に関する。

【0002】

【従来の技術】コンピュータイメージ生成システムでは、テクスチャマッピングを使用して、表示すべきオブジェクト又は特徴についてのリアルなイメージの再生を得ることができる。テクスチャマッピングは、イメージをオブジェクトのサーフェスに適用する手法である。テクスチャマッピングを適用することにより得られる一般的な利点は、リアル性を加えることである。テクスチャマッピングをおこなうのに、数多くの方法が用いられてきた。どの方法を選択するかとは無関係に、テクスチャマップを作成するときには、マトリックスに一定数のセルがある。マトリックスにおけるセル数（例えば、 256×256 ）は、テクスチャマップの解像度である。解像度は、詳細レベルとして知られている一定の所定見掛け距離でのオブジェクトを表す。この「見掛け距離」は、視界のサイズ、オブジェクトまでのビューワの距離及びオブジェクトの方位の関数である。オブジェクトがビューワから相対的な距離を動くと、テクスチャリングすべきオブジェクト内の個々のピクセルに寄与するテクスチャマップにおけるセルの全てを含ませるのが計算上やっかいとなるので、テクスチャカラー値の決定において問題が生じる。この問題を解決するには、オブジェクトからビューワまでの異なる相対距離で使用する同じテクスチャパターンの一連のテクスチャマップを保存する。この一連のテクスチャマップは、「ズームピラミッド」と称されることがある。ズームピラミッドは、高解像度テクスチャマップから開始し、ピラミッドにおける前のマップの解像度がもっと低いバージョンである連続したマップを含む。ズームピラミッドは、効果的に表示すべきシーンを複数の所定範囲の間隔に分割する。ここで、各範囲の間隔は、異なるそれぞれの詳細レベル（LOD）に相当する。オブジェクトからビューワまでの相対距離を異ならせたテクスチャマップを作成することは、複数のテクスチャ詳細レベル（LOD）を確定することとして知られている。LOD作成するには、最高解像度のテクスチャであるであろうベースLODを確定することによりLODを作成する。換言すれば、所定領域での最大数及び最小サイズのセルを確定することである。最高解像度テクスチャマップは、観察者に最も近く、オブジェクトがビューワに近いときにテクスチャをオブジェクトに適用するのに使用される。より低い解像度のバージョンを配置して、同じ領域についての相当するより少数で、より大きなサイズのセルでそれぞれの減

少量の詳細を含ませる。より低い解像度のバージョンは、前のマップからのセルデータをフィルタリングすることにより作成し、オブジェクトがビューワからさらに動くときにテクスチャを適用するのに使用する。テクスチャマッピングプロセスでは、オブジェクトがビューワに対して現れる距離又は見掛け距離に基づいて、ズームピラミッドにおいてどのテクスチャマップを使用するかについて、決定しなければならない。LOD計算により、この決定がなされる。LODの計算は、テクスチャセルとピクセルサイズの比較に基づいておこなわれ、最も近いセルとピクセルサイズとが一致する2つの連続テクスチャマップを選択する。これらの2つのマップの解像度は、テクスチャリングされるオブジェクトをブラケットするような解像度である。これらの2つのテクスチャLODマップの各々からのデータは、メモリーから出す。各マップからのデータは、双一次内挿等の2D重み付き関数を用いてフィルタリングする。これらの2つのテクスチャLODマップからのフィルタリングしたデータを、次に、2つの選択したテクスチャLOD間のオブジェクトのロケーションに依存した線形内挿によりブレンドされる。LODブレンドにより、オブジェクトが2つの選択されたテクスチャLODについて設定された距離間のシーンにおいて動いているときにテクスチャLOD間の遷移が円滑となる。2つの連続マップと線形内挿におけるこのフィルタリング方法は、一般的に「三線内挿」と称せられ、米国特許第4,727,365号に記載されている。この方法を用いてデータを出すには、8回のピクセルアクセスと、一つのスクリーンピクセル当たり7〜13マルチプライに伴う一定の関連コストがある。

【0003】計算をセーブするのにたびたび使用されるもっと簡単な方法で必要なことは、ズームテクスチャピラミッドのより近いLODマップであるLOD_Nにおけるテクセルをフィルタリングすることだけである。より近いマップからの双一次フィルタリングテクスチャ値は、4回のピクセルアクセスと、3線内挿として1ピクセル当たり半分未満のマルチプライを必要としているだけである。ポリゴンまでの見掛け距離が増加するにつれて、ピクセル値を得るのに使用されるLODマップは、あるあらかじめ確定したしきい値距離でLOD_NからLOD_{N+1}にジャンプする。この点で、フィルタリングが、LOD_{N+1}で専ら生じる。この簡略化された手法により、3線内挿で必要とされるような2つのLODレベル間のブレンド工程を排除できる。しかしながら、この方法は、3線内挿よりも計算的にもっと効率的であるが、見掛け距離の変化に伴い、ほとんどのテクスチャマップにとって望ましくない可視アーチファクトを生じる。その結果、関連した望ましくない可視アーチファクトを生じないテクスチャマップフィルタリングの簡略化された計算的に安価な方法が必要とされてい

る。

【0004】

【発明が解決しようとする課題】本発明の目的は、計算的に効率的であり、且つそれによりテクスチャメモリーバンド幅要件を減少するとともに、同時に、従来法よりも可視アーチファクトを顕著に減少する、LOD選択方法を提供することである。

【0005】

【課題を解決するための手段】本発明によれば、3線内挿に使用される2つのフィルタリングしたLOD間のより高価で正確な線形ブレンディングを必要とすることなく、ポリゴンを介するテクスチャの詳細レベル（LOD）遷移の可視効果を最小限とする方法が提供される。本発明の方法は、1）グラフィックス要素（例えば、ポリゴン）によりカバーされる各ピクセルについてのテクスチャLOD値を計算する工程と、2）ディザリングオフセット値を各LOD計算値に適用してディザードピクセルLOD値を得る工程と、3）ディザードピクセルLOD値により同定されるLODテクスチャマップからテクスチャデータを得る工程と、を含んでなることを特徴とする方法が提供される。

【0006】具体的な実施態様においては、スクリーンスペースは、複数のスパンを含んでなる。各スパンは、例えば、寸法4×4のピクセルグリッドをさらに含んでなる。本発明の方法では、スパンにおけるポリゴンによりカバーされる各ピクセルについてのLOD値を計算する。本発明の方法を、次のスパンにおけるこれらのカバーされたピクセルに適用する。スパン処理に関するさらなる詳細な記載が、米国特許第4,811,245号にある。本方法では、ルックアップテーブル又はそれと同等の保存装置に保存できるディザード値のマトリックスをさらに作成する必要がある。

【0007】LODの計算値にディザリングオフセットを適用することにより、あるテクスチャピクセル計算値を遠テクスチャLODマップにプッシュ（ $LOD=N+1$ ）し、他のピクセルテクスチャ値を近LODマップにプル（ $LOD=N$ ）する可能性がある。本発明の方法によれば、ディザードオフセット値を、加算、乗算等により、ピクセルLOD計算値に適用する。丸め又は切り捨て後の適用による整数の結果により、フィルタリングについての近又は遠テクスチャマップが同定される。一定のLODピクセル計算値の百分率は、ディザリングオフセットの加算によっては影響されず、他のものは、いずれかのディザードオフセットの適用の前に選択されるものとは異なるマップに示される。もしディザリングオフセット値が負であるならば、選択されたLODマップを近マップにプルバックするように変化する。もしディザリングオフセットが正数であるならば、LOD選択における結果が遠マップまでプッシュされるように変化する。

【0008】ここで、一定の用語を定義するのが適当で

ある。ディザリングオフセット値は、通常のピクセルLOD計算値に適用される値である。ディザリングオフセット範囲は、本方法で使用するために保存されるディザリングオフセット値のマトリックスの最小値及び最大値により設定される。ディザ範囲は、最小値を最大値から差し引くことにより得られる値である。ディザードLOD範囲は、選択LODが本発明のディザリング法より変更されるLOD計算値範囲である。

【0009】ディザリングオフセット範囲は、上記したように、影響されるLODテクスチャ計算値の範囲であるディザードLOD範囲を決定する。しかしながら、一定用途では、一定ディザリングオフセット範囲のみで、許容できる結果が得られることがある。本方法の有効性の一つの可能性のある本質的な尺度は、ピクセルアレイ（例えば、スパン）を介したディザード結果を統合する眼に依存する。範囲が広すぎるか狭すぎると、関連する望ましくないビジュアルアーチファクトを示す。もし、例えば、ディザリングオフセット値は、（0～.75）又は（0～1）の範囲であるならば、眼が2つの選択された隣接するテクスチャマップを2つの重ね合ったイメージとして認識することがある点で、望ましくないビジュアルアーチファクトが生じる。小さいディザリングオフセット範囲、例えば、0～.2について、ディザードLOD範囲は小さく、眼は、2つの隣接するLODマップ間のジャンプを見掛け距離変化として認識することがある。4～.6のディザ範囲を求めて、ほとんどの用途に最適な結果を得ることができたが、どの範囲が最適かは、用途による。

【0010】本発明の別の実施態様においては、LOD計算値に適用されるディザリングオフセットの量を、制御状態変数設定により変調する。変調は、ユーザーにより選択され、別個の変調工程を定義することができる。変調により、ディザリングオフセット値を減衰又は増幅できる。例えば、減衰器により、半分、4分の1、8分の1及び減衰なしの工程でディザリングオフセットを減少できる。適用される減衰は、ポリゴン当たり、テクスチャ当たり、又はこれらの両方について修正できる。減衰度の修正により、ディザリングオフセット範囲が対応して影響される。より詳細には、減衰量を増加するにつれて、影響されるピクセルオフセット値のより狭い範囲にトランスレートする対応する最大ディザリングオフセットが減少する。

【0011】計算のセーブは、本方法により、ズームテクスチャピラミッドのより近いLODテクスチャマップのみにおいて各ピクセルをフィルタリングすることにより実現できる。これにより、3線内挿に必要とされる2つの隣接するLODにおけるフィルタリングされた結果間のブレンディング工程がなくなる。

【0012】

【発明の実施の形態】本発明を特徴付ける新規性の種々

の特徴を、添付し且つ開示の一部を形成する独自性について示す。本発明のよりよい理解、その具体的利点及びその使用により得られる具体的目的については、図面及び、本発明の好ましい実施態様に説明及び記載されている記載事項を参照すべきである。

【0013】本発明は、ディザリングオフセットをテクスチャLOD計算値に適用して、ポリゴンを介するテクスチャ詳細レベル（LOD）遷移の可視効果を、3線内挿に使用されるような2つのフィルタリングされたLOD間のもっと高価で且つ正確な線形ブレンディングを必要とせずに、最小限とする方法に関する。ディザリングオフセットは、典型的には、スクリーン（ディスプレイ）スペースにおけるピクセルの位置に基づいて個々のピクセルについて適用される値によって表される。

【0014】図1は、テクスチャをポリゴンのサーフェスに適用するピクセルのLOD値を計算する方法を図示したものである。本方法は、サーフェス13により表されるオブジェクトまでのビュー10の見掛け距離を求める工程を含んでなる。ピクセルのサイズを、テクスチャマップのズームピラミッドにおけるテクスチャセル又はテクセルのサイズとを比較することにより、矢印16により表される見掛け距離を、表面13がディスプレイスクリーン上をおおう各ピクセルについて計算する。図1は、ズームピラミッドに射影された後に現れるピクセル15を示す。射影されたピクセル15は、中心11を有する。2つのLODテクスチャマップ12及び14は、それぞれLOD_N及びLOD_{N+1}を表す。LOD14は、その解像度に基づく1サイズのテクセル18を有し、LOD12は、LOD14よりも低い解像度を有するものに基づくより大きなサイズのテクセル19を有する。ピクセル15については、ピクセルサイズ／テクセルサイズの比較により、サイズが最も近くに対応する2つのマップの間にあるピクセルを求める。比較の結果は、整数と端数とからなる数、例えば、3.2である。この数は、オブジェクト13のビュー10までの見掛け距離16を表す。その後、3線内挿について、射影17a及び17bを包囲する各それぞれのマップ内の4つのテクセルに相当するテクスチャ値の内挿を実施して、各マップにおいて単一のテクスチャ値を生じる。各マップ内の内挿は、双一次内挿を含むいずれの2D内挿法でもよい。2つの計算値を、次に、2つの結果の間の1D内挿に対する入力として利用する。この方法は、コンピュータ生成システムにおいてポリゴンをテクスチャリングする最もリアルな方法であることを示している。この方法は、極めて正確ではあるが、PCS用3D表現コンピュータチップを設計するという今日の所要要件を満たすには計算上高価である。

【0015】図2は、本発明を構成する一般的な方法の工程を示すフローチャートである。この方法は、オブジェクトによりカバーされるピクセルのテクスチャマップ

ングをイネープリングすることにより、ステップ20から開始する。ステップ22では、テクスチャLOD値を、意図するピクセルについて計算し、整数成分であるintLODと端数成分であるfracLODとからなるものとして表すことができる（すなわち、LOD=intLOD, fracLOD）。整数成分は、使用される最も近い（ビューワから）LODマップであるLOD=Nを表し、一方、端数成分は、近テクスチャマップと遠テクスチャマップとの間のピクセルLOD値の相対距離を表す。端数LOD成分であるfracLODが半分（1/2）未満であるときにはいつでも、ディザリングオフセットの加算からの可能性のある寄与の前に、近テクスチャLODマップであるLOD=Nを選択して、テクスチャ値を得る。端数LODが半分以上を超える時には、遠テクスチャLODマップであるLOD=N+1を選択してテクスチャ値を取り出す。

【0016】ステップ24で、ディザリングオフセットを計算する。好ましい実施態様では、まず、意図するピクセルのディスプレイ（I、J）座標を求め、（I、J）アドレスをインデックスとして使用して、コンピュータで計算して、ディザオフセット値を得るまでのテーブルルックアップとする。別法として、結合論理を使用して、ピクセルのロケーションに基づくディザリングオフセット値を計算できる。ディザリングオフセット計算は、もしポリゴンがスパン処理を用いて表現されているならば、ピクセルスパンにおけるピクセルのロケーションに基づくことができる。ステップ26では、ディザリングオフセット計算値を、ステップ24からのLOD計算値に適用し、その結果として、ディザードピクセルLOD値が得られる。また、ディザードピクセルLOD値は、整数値と端数値の形態である。次に、ステップ28において、.5以上で高LOD数が選択され、.5未満で低LOD値が選択される丸めにより、LODの選択をおこなう。また、LODを整数値のみに基づいて選択する場合には、切り捨てを使用することができる。ディザリングオフセットを適用することにより、あるピクセルのテクスチャLOD計算値を、さもなければ遠マップから選択されたであろう近整数テクスチャマップ値であるLOD=Nに「プル（pulling）」する効果がある。同様に、オフセットにより、他のLOD計算値を、さもなければ近マップを選択したであろう遠テクスチャLODマップであるLOD=N+1にプッシュしてもよい。丸めを使用するときには、ステップ26の結果の端数部が近テクスチャマップと遠テクスチャマップとの間の距離の2分の1未満である場合に近テクスチャマップへのプルが生じる。ステップ26の結果の端数部が2つのテクスチャマップ間の距離の2分の1に等しいかそれを超える場合には、遠マップへのプッシュが生じる。

【0017】図3に、例として、選択ディザ範囲が、4である場合、ピクセルLOD計算値を近テクスチャマッ

プである $LOD=N$ か、遠テクスチャマップである $LOD=N+1$ に割り当てられる確率を示す。ディザ範囲が、4を有する数多くのディザリングオフセット範囲がある。一つのこのような範囲が $(-1, 2 \sim +1, 2)$ であり、別の範囲が $(0 \sim 4)$ である。図3のグラフは、ディザリングオフセット範囲が $(-1, 2 \sim +1, 2)$ であり、丸めを使用する確率を示す。切り捨てを使用する場合には、もしバイアス、5をディザリングオフセットに加えるならば、グラフをまだ適用する。バイアス値の適用を、図7を参照して説明する。さらに、バイアスを加えるならば、ディザリングオフセット範囲 $(0 \sim 4)$ についてグラフを適用する。丸めの場合、バイアスは $(-1, 2)$ でなければならず、切り捨ての場合、バイアスは $(+1, 3)$ でなければならない。

【0018】この例において（オフセット範囲 $-1, 2 \sim +1, 2$ ）、ディザリングオフセットをピクセル LOD 計算値に加える。ディザオフセットの加算の前に LOD 計算値が $N+1, 3$ 未満の全てのピクセルが自動的に確率100%で近テクスチャマップである $LOD=N$ に割り当てられる。ディザリングオフセットは、ディザ範囲、4が所与のものとするれば、これらの計算値には影響がない。 LOD 計算値が $N+1, 7$ 以上である全てのピクセルは、遠テクスチャマップである $LOD=N+1$ に自動的に割り当てられるであろう。ここでも、ディザリングオフセットは、これらのピクセル LOD 計算値には影響を及ぼさない。 LOD 計算値が $(N+1, 3 \sim N+1, 7)$ 未満の範囲にあるピクセルのみが、ディザリングオフセットの加算ゆえに、異なる LOD マップに割り当てられる何らかの確率がある。例えば、 LOD 計算値が $N+1, 5$ のピクセルは、 $LOD=N$ 又は $LOD=N+1$ に割り当てられる確率が50%である。ディザ範囲を広げると、マップ N 又は $N+1$ の選択の候補としてのピクセル LOD 計算値のより大きな百分率に影響することがわかる。 $N+1, 3 \sim +1, 7$ 未満の狭い範囲における LOD 計算値のみがプッシュ又はプル候補である図示されている例において、全ての他の値が、ディザリングオフセットの寄与とは無関係に直ちに特定のマップに割り当てられる。

【0019】図4は、ディザ範囲1についてピクセル LOD 計算値が「プッシュ」又は「プル」される確率を示す。ここでも、ディザ範囲は、数多くのオフセット範囲、例えば $(-1, 5 \sim +1, 5)$ 又は $(0 \sim 1)$ 、により得ることができる。グラフは、丸めを使用する場合のディザリング値 $(-1, 5 \sim +1, 5)$ の範囲、又は切り捨てを使用する場合にはこの同様の範囲に $+1, 5$ のバイアスを加算したものについての確率を示す。 $(0 \sim 1)$ の範囲の場合、切り捨てを使用するならば図4のグラフを適用し、丸めを使用するならばバイアス $(-1, 5)$ を加算しなければならない。この例において、全てのピクセル LOD 計算値についての LOD テクスチャ計算値が、近

テクスチャマップへのプル候補又は遠テクスチャマップへのプッシュ候補であろう。図3において範囲で示したような自動割り当ての等価領域はない。したがって、図3及び図4に示した例から、ディザリングオフセットの範囲を調整することにより、 LOD テクスチャ計算値の領域を制御できることが明らかである。

【0020】図5は、遠近法で示した角形ポリゴンについてのディザ範囲、4の LOD 間の遷移を示す。当業者に周知のように、見掛け距離は、ビューワからポリゴンまでの距離、及び観察者からポリゴンまでのビューレイに関するポリゴンの方位の関数である。ポリゴンの付形遷移領域におけるピクセルは、隣接整数 LOD のディザブレンドである。ポリゴンの白色領域におけるピクセルの全てを、最も近い整数 LOD に割り当てられる。

【0021】図6は、3つの代表的なディザリング範囲についてのピクセル計算値に対するディザリングオフセットの寄与を表すグラフである：1) 完全ディザリングの場合（対角線）—ディザ範囲1 $(-1, 5 \sim +1, 5)$ 、2) ディザリングなし [実線階段ステップ] —ディザ範囲0 $(0 \sim 0)$ 、及び中間の場合3) 部分ディザリング [点線] —ディザ範囲、25 $(-1, 125 \sim +1, 125)$ 。最初の場合である完全ディザリングは、各 LOD 計算値が近又は遠テクスチャマップへのプル又はプッシュされる候補である真の3線内挿モードを生じるものに等しい。上記したように、これにより、望ましくないアーチファクトを生じる場合がある。第二の場合である実線階段線は、ディザリングなしの場合の影響を示している。この場合、ピクセル計算値は、あるあらかじめ定めたいきい値、典型的にはオブジェクトをブラケットする2つの LOD マップ間の距離の半分で、一つの LOD マップから次の LOD マップに急激に遷移又はジャンプする。点線により表される第三の場合は、ディザ範囲、25を表す。この場合、ディザオフセットの加算は、ピクセル LOD 計算値の3/4に影響を及ぼさない。より詳細には、本方法は、ディザリングオフセットの加算の前の端数成分が、75を超えるかそれに等しいピクセルオフセット計算値を生じないであろう。

【0022】図7は、図形要素によりカバーされる各ピクセルについての詳細レベル（ LOD ）テクスチャマップを選択するディザリングユニット装置70の一般的な成分を示すブロック図である。この装置は、入力として、テクスチャであるピクセルのスクリーンスペースにおける座標を必要とする。例えば、テクスチャリングされるべきピクセルのI及びJスクリーン座標の2つの最も有意性の小さいビット71を使用して、ピクセルのロケーションを表すことができる。これらの2つの値をインデックスとして、二次元ディザパターン生成器72に入力する。生成器72は、ディザリングオフセットのパターン又はアレイを含むテーブルルックアップの形態をとることができる。別法として、生成器72は、異なる

テクスチャ又はポリゴンのオフセットを生成する組み合わせ論理を含んでなることができる。パターン生成器は、スクリーンスペースにおける特定のピクセルのアドレスビットに基づく単一ディザオフセット値を戻す役割を果たす。

【0023】例えば、一実施態様においては、テーブルルックアップを、一度に一スパンのポリゴンを処理するシステムに使用する。テーブルルックアップ72をオーガナイズして、4x4アレイ等のスパンコンフィギュレーションをミラーリングすることにより、スパンの物理的寸法を反映する。テーブル72により取り出したディザリングオフセットを、テーブル72に対するインデックスとしての現在のスパンにおけるピクセルの位置を利用することにより求める。

【0024】テーブルルックアップ72から、単一ディザオフセット値73を出力し、ディザ値変調器74への入力として供給する。変調器74は、必要に応じて設けられる。変調器74により、いくつかのディザリングオフセット範囲がルックアップに保存された一組の値のみから得ることができる。ディザ値73に適用する変調度は、制御状態変数77としてのディザ値変調器74への第二の入力により決定される。変調器74は、保存オフセット値を増加又は減少するようにプログラム化できる。例えば、変調器74は、減衰器であることができる。この場合、制御状態変数77を設定して、範囲ゼロにおけるディザリングオフセットを完全減衰に適用するようにできる。ここで、中間値1、2及び3は、それぞれ2分の1減衰、4分の1減衰及び8分の1減衰に相当する。例えば、制御状態変数77は、8シフトを提供できるが5のみを使用するエンコードされる3ビットであることができる。ディザリングオフセット値73は、4ビット数である。次に、減衰器により、この数が、減衰量だけ右にシフトされ、リードビットに0を挿入する。減衰は、0シフト（全ウェイト）、1シフト（半分）、2シフト（4分の1）、3シフト（8分の1）又は4シフト（完全シフト、したがってディセーブルされた）であることができる。

【0025】次に、変調ディザ値75を、ピクセルLOD計算値78と一緒に加算器76に入力して、ディザードピクセルLOD値82を得る。図7に、任意のバイアス手段を示す。この実施態様では、LODバイアス状態変数79も、加算器76に入力する。バイアス状態値を入力することにより、最終加算結果の切り捨てをコンピュータ計算の面より高価な丸め操作（それが無い場合、同様の結果を得るのに実施しなければならない）の代わりにできる。また、バイアス状態値を使用して、一つのLODから別のLODに遷移する場所を変更してもよい。

【0026】加算器76は、整数と端数成分を有するディザードピクセルLOD値82を出力する。次に、ディ

ザードピクセルLOD値を、手段80に入力して、切り捨て又は丸めの数学的オペレーションを実施して、整数のみを得る。ディザードピクセルLOD値の整数部81を出力して、LODマップを選択する。

【0027】図8は、二次元テクスチャディザパターンテーブルの内部コンフィギュレーションを示したものである。ルックアップテーブル90の寸法をスパン（4x4）の寸法に整合させて、スクリーンスペースにおけるピクセルのI及びJアドレスの2つの最も優位性の小さいビットをテーブルに加えることにより、特定のピクセルのルックアップテーブルからディザオフセット値を選択する。具体的実施態様において、単一のルックアップテーブルを、各処理スパンについて使用する。しかしながら、別の実施態様で、ディスプレイスペースにおける各及び全てのピクセルのディザオフセットを作成できる。テーブルルックアップ90に保存されたディザードオフセット値92は、ゼロ〜15/16の範囲の4ビット値である。具体的実施態様において、これらの16の値を選択して、選択されたディザ範囲に均一に分布するが、本方法を、いずれかのユーザーの選択した値の分布（一定の分布は他のものよりも視覚的にアピールする）を反映するように実行できる。

【0028】以上、本発明のいくつかの実施態様と変更態様を詳細に説明したが、本発明の開示及び教示は、当業者に数多くの代替構成を示唆していること、及び本発明の範囲は、請求の範囲によってのみ限定されることは明らかであろう。

【図面の簡単な説明】

【図1】従来技術による3線内挿フィルタリング法を示す。

【図2】単一ピクセルについての本発明の方法の工程を示す。

【図3】ディザ範囲、4についてのディザリングの結果を示すグラフである。

【図4】ディザ範囲1についてのディザリングの結果を示すグラフである。

【図5】ポリゴンについてのディザリング範囲、4を示す斜視図である。

【図6】ディザードピクセルLOD値とピクセルLOD計算値との関係を示すグラフである。

【図7】本発明のハードウェアの実施態様のブロック図である。

【図8】テクスチャディザテーブルマトリックスの一実施態様である。

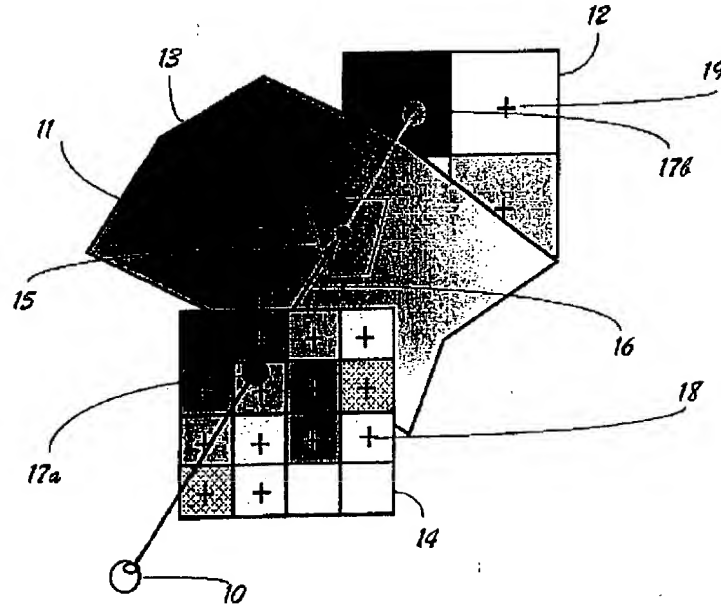
【符号の説明】

- 10 ビューワ
- 11 中心
- 12、14 LODテクスチャマップ
- 13 サーフェス
- 15 ピクセル

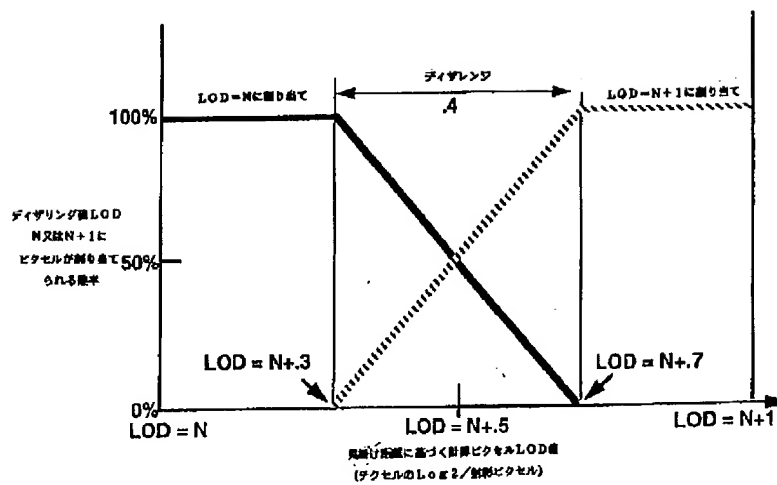
16 見掛け距離
 17a、17b 射影
 18、19 テクセル
 70 デザリングユニット装置
 71 ビット
 72 二次元デザパターン生成器

73 デザリングオフセット値
 74 デザ値変調器
 76 加算器
 77 制御状態変数
 81 デザードピクセルLOD値の整数部

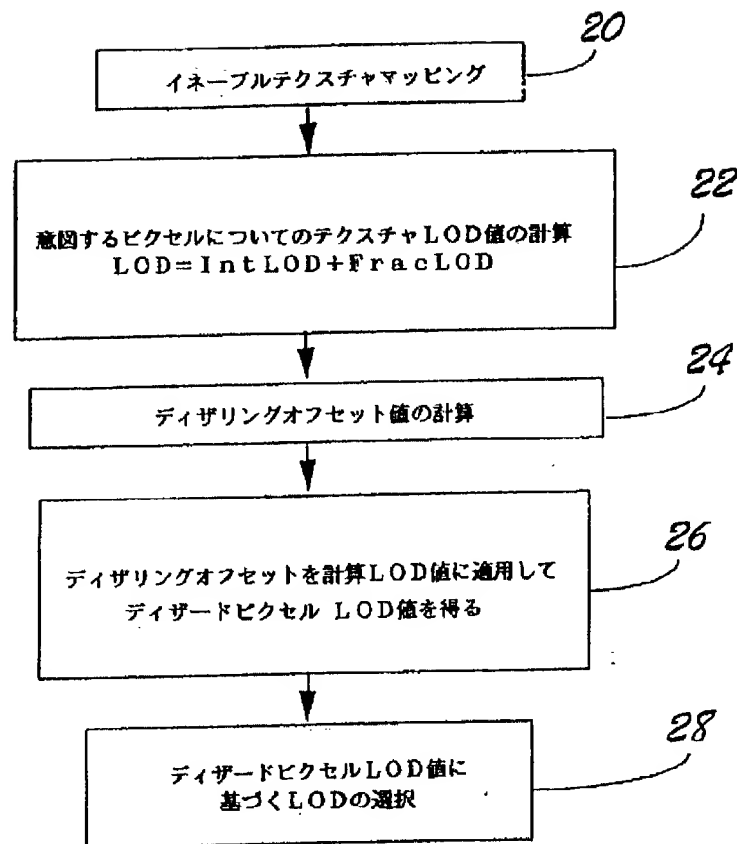
【図1】



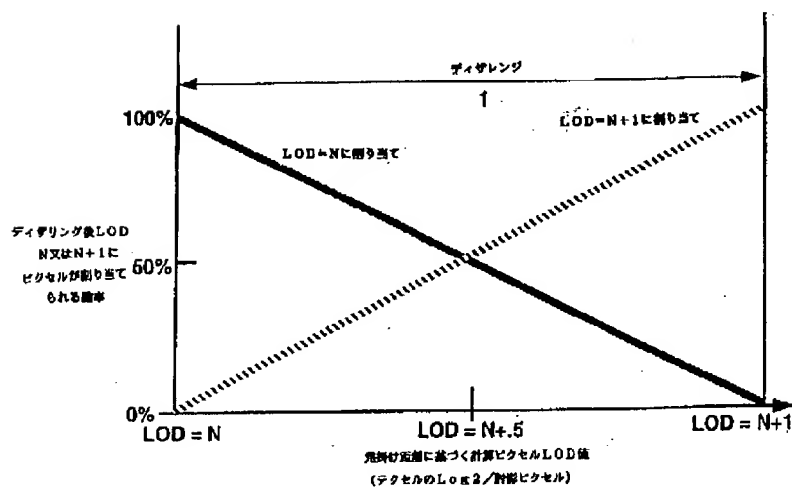
【図3】



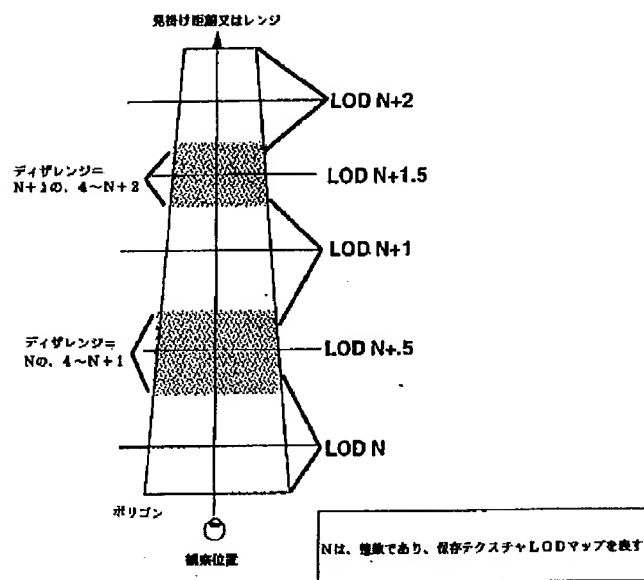
【図2】



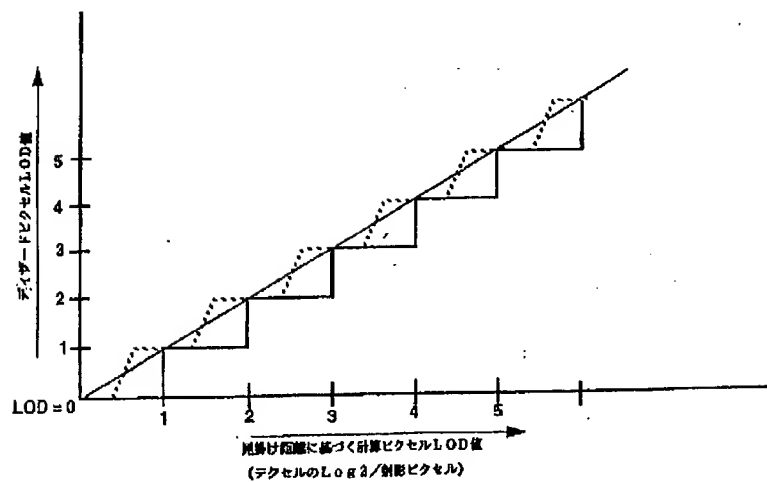
【図4】



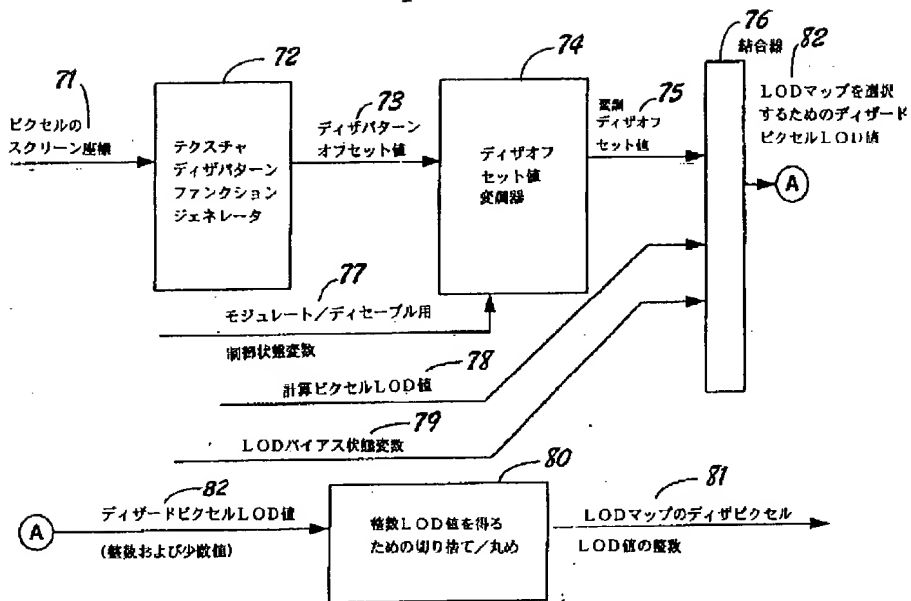
【図 5】



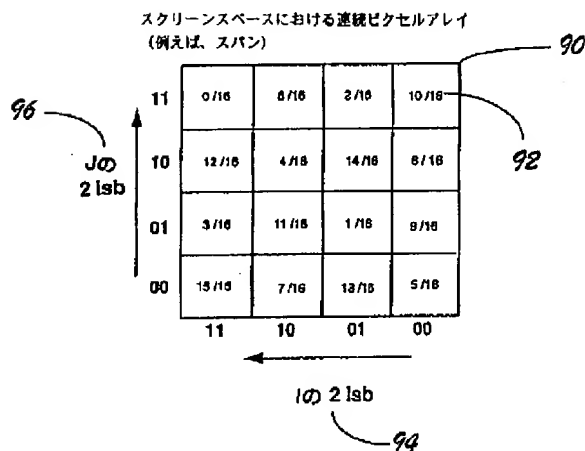
【図 6】



【図7】



【図8】



フロントページの続き

(72)発明者 マイケル マンター
アメリカ合衆国、フロリダ州 32829、オー
ーランド、サドル クリーク プレイス
4318

(72)発明者 ラルフ クレイトン テイラー
アメリカ合衆国、フロリダ州 32724、デ
イランド、ロックウェル ハイ츠 1548
(72)発明者 スティーブン マンノ
アメリカ合衆国、フロリダ州 32174、オー
ーモンド ビーチ、サンド クリーク レ
ーン 424

【外国語明細書】

11103.EWG
ag\F:\WORK\1172\11103\SPEC\11103.ewg

5

METHOD AND APPARATUS FOR
TEXTURE LEVEL OF DETAIL DITHERING

10 The present invention relates generally to
real time computer image generation systems, and, more
particularly, to means and method for texture map
filtering.

15 In computer image generation systems,
texture mapping may be used to achieve realistic image
reproduction for objects or features to be displayed.
Texture mapping is a technique that applies an image
to an object's surface. The general benefit provided
by the application of texture mapping is to add
realism. Many methods have been employed to accomplish
texture mapping. Irrespective of which method one
20 selects, when a texture map is created it will have a
certain number of cells in the matrix. The number of
cells in the matrix (e.g. 256x256) , is the resolution
of the texture map. The resolution represents the
object at a certain predetermined apparent distance
25 which is known as a level of detail. This "apparent
distance" is a function of the size of the field of
view, the distance of the viewer to the object and the
orientation of the object. As an object moves in
relative distance from the viewer a problem occurs in
30 determining the texture color value in that it becomes
computationally burdensome to include all of the cells
in the texture map that contribute to the individual
pixels within the object to be textured. The solution
is to store a sequence of texture maps of the same
35 texture pattern to be used at different relative
distances from the object to the viewer. This sequence

-2-

of texture maps is sometimes called a zoom pyramid. The zoom pyramid starts with a high resolution texture map and includes successive maps that are lower resolution versions of the previous map in the pyramid. The zoom pyramid effectively divides the scene to be displayed into a plurality of predetermined range intervals, where each range interval corresponds to a different respective level of detail (LOD). Creating texture maps at different relative distances from the object to the viewer is known as establishing multiple texture levels of detail (LODs). The manner in which the LODs are created is by establishing a base LOD which will be the texture map with the highest resolution, in other words the greatest number and smallest sized cells for a predetermined area. The highest resolution texture map is closest to the observer and is used for applying the texture to an object when the object is near to the viewer. Lower resolution versions are arranged to contain a respective decreasing amount of detail with correspondingly smaller number and larger sized cells for the same area. The lower resolution versions are created by filtering the cell data from the previous map and are used for applying texture as the object moves further from the viewer. In the texture mapping process, a determination must be made of which texture maps in the zoom pyramid to use based on the distance the object appears to be to the viewer or apparent distance. The LOD calculation makes this

-3-

determination. The LOD calculation is based on a texture cell to pixel size comparison and selects the two consecutive texture maps that have the nearest cell to pixel size correspondence. The resolution of these two maps are of such resolution such that they bracket the object to be textured. Data from each of these two texture LOD maps is retrieved from memory. The data from each map is filtered using a 2D weighting function such as bilinear interpolation. The filtered data from these two texture LOD maps is then blended by linear interpolation depending on the location of the object between the two selected texture LODs. The LOD blending provides a smooth transition between texture LODs when the object is moving in the scene between the distances set for the two selected texture LODs. This technique of filtering in two successive maps and a linear interpolation is commonly referred to as trilinear interpolation and is described in U.S. Patent 4,727,365. Retrieving data with this method has a constant associated cost of eight pixel accesses and between seven and thirteen multiplies per screen pixel.

A simpler method that is often used to save computations, is to only filter the texels in the closer LOD map, LOD N, of the zoom texture pyramid. The bilinearly filtered texture value from the closer map would require only four pixels accesses and less than one half the multiplies per pixel as trilinear interpolation. As the apparent distance to the polygon

- 4 -

increases, the LOD map used to obtain the pixel value will jump from LOD N to LOD N+1 at some pre-established threshold distance, at which point the filtering will occur exclusively in LOD N+1. This
5 simplified approach eliminates the step of blending between the two LOD levels, as required in trilinear interpolation. This method, however, while computationally more efficient than trilinear
10 interpolation, produces undesirable visible artifacts for most texture maps as the apparent distance changes. As a consequence, there exists a need for a simplified, computationally inexpensive method of texture map filtering that does not produce associated undesirable visible artifacts.

15 The present invention is directed to a method of LOD selection which is computationally efficient and, thereby reduces texture memory bandwidth requirements, while at the same time significantly reduces the visible artifacts over
20 conventional methods.

In accordance with the present invention, there is provided a method for minimizing the visible effects of texture Level of Detail (LOD) transitions across a polygon without the need for the more
25 expensive and accurate linear blending between two filtered LODs used in trilinear interpolation. The method of the present invention comprises the steps of:
: 1) computing a texture LOD value for each pixel covered by a graphics primitive (e.g. polygon), 2)

-5-

applying a dithering offset value to each computed LOD value resulting in a dithered pixel LOD value, and 3) obtaining texture data from the LOD texture map identified by the dithered pixel LOD value.

5 In an illustrative embodiment, screen space is comprised of a plurality of spans, wherein each span is further comprised of a grid of pixels, for example, of dimension 4x4. The method of the present invention calculates an LOD value for each pixel covered by a
10 polygon in a span. The method will be applied to those covered pixels in a subsequent span. Further details regarding span processing can be found in U.S. Patent 4,811,245. The method further requires the creation
15 of a matrix of dither values, which may be stored in a look-up table or equivalent storage device.

 Application of a dithering offset to the computed LOD value has the potential effect of pushing some computed texture pixel values to the far texture LOD map ($LOD=N+1$), and pulling other pixel texture
20 values to the near LOD map ($LOD=N$). Pursuant to the method of the invention, a dithered offset value is applied, such as by adding, multiplying, etc., to the computed pixel LOD value. The integer result of the application, after roundoff or truncation, identifies
25 either the near or far texture map for filtering. A certain percentage of computed LOD pixel values will be unaffected by the addition of the dithering offset while others will point to a map different than what would have been selected prior to any dither offset

-6-

application. If a dithering offset value is negative, the change will be to pull the selected LOD map back to the near map. If the dithering offset is a positive number the change will result in the LOD selection being pushed up to the far map.

It is appropriate here to define certain terms. The dithering offset values are the values that are applied to the conventional computed pixel LOD value. The dithering offset range is set by the minimum and maximum values of the matrix of dithering offset values that are stored for use in the method. The dither range is the value obtained by subtracting the minimum value from the maximum value. The dithered LOD range is the range of computed LOD values in which the selected LOD is changed due to the dithering method of the present invention.

The dithering offset range will determine the dithered LOD range which as noted above is the range of computed LOD texture values affected. It is observed, however, that for certain applications, only a certain dithering offset range will yield acceptable results. One possible subjective measure of the effectiveness of the present method depends on the eye integrating the dithered result across the pixel array (e.g., span). Ranges which are either too wide or too narrow will have associated undesirable visual artifacts. If, for example, the values of the dithering offsets are in the range of (0 to .75) or (0 to 1) an undesirable visual artifact results in that

-7-

the eye will sometimes perceive the selection of two adjacent texture maps as two superimposed images. For small dithering offset ranges, such as 0 to .2, the dithered LOD range will be small and the eye will often perceive a jumping between two adjacent LOD maps as the apparent distance changes. Dither ranges from .4 to .6 have been determined to provide the best results for most applications, however, what range will be best will be application specific.

10 In another embodiment of the present invention, the amount of dithering offset applied to the computed LOD value is modulated according to a control state variable setting. The modulation is user selectable and can define discrete modulation steps.

15 The modulation may attenuate or amplify the dithering offset values. For example, an attenuator may reduce the dithering offset in steps of : half, quarter, eighth and no attenuation. The applied attenuation can be modified either per polygon, per texture, or both.

20 Through modification of the degree of attenuation the dithering offset range is correspondingly affected. More specifically, as increasing amounts of attenuation are applied, the corresponding maximum dithering offset is reduced which translates to a

25 narrower range of affected pixel offset values.

A computational savings is realized by the present method by filtering each pixel in only the closer LOD texture map of a zoom texture pyramid. This eliminates the step of blending between the filtered

- 8 -

results in the two adjacent LODs that are required for trilinear interpolation.

The various features of novelty which characterize the invention are pointed out with particularity in the claims annexed to and forming a part of the disclosure. For a better understanding of the invention, its operating advantages, and specific objects attained by its use, reference should be made to the drawings and descriptive matter of which there are illustrated and described preferred embodiments of the invention.

FIG. 1 illustrates a trilinear interpolation filtering method of the prior art.

FIG. 2 illustrates the method steps of the present invention for a single pixel.

FIG. 3 graphically illustrates the result of dithering for a dither range of .4.

FIG. 4 graphically illustrates the result of dithering for a dither range of 1.

FIG. 5 illustrates a dithering range of .4 for a polygon in perspective view.

FIG. 6 graphically illustrates the dithered pixel LOD value as a function of the computed pixel LOD value.

FIG. 7 illustrates a block diagram of a hardware embodiment of the present invention.

FIG. 8 illustrates one embodiment of a texture dither table matrix.

-9-

The present invention is directed to a method for applying dithering offsets to computed texture LOD values to minimize the visible effects of texture Level of Detail (LOD) transitions across a polygon without the need for the more expensive and accurate linear blending between two filtered LODs as used in trilinear interpolation. The dithering offsets are typically represented by values which are applied for individual pixels based on the pixel's position in screen (display) space.

Referring now to FIG. 1, there is shown pictorial representation of the method of computing an LOD value for a pixel for applying texture to the surface of a polygon. The method comprises the steps of determining an apparent distance of the viewer to the object represented by surface 13. The apparent distance, represented by arrow 16 is computed for each pixel that the surface 13 covers on the display screen by comparing the size of the pixel to the size of the texture cells or texels in the zoom pyramid of texture maps. Figure 1 shows a pixel 15 as it appears after being projected into the zoom pyramid. The projected pixel 15 has center 11. Two LOD texture maps 12 and 14 represent LOD N and LOD N+1, respectively. LOD 14 has texels 18 of one size based on its resolution and LOD 12 has texels 19 of a larger size based on it having a lower resolution than LOD 14. For pixel 15 the pixel size/texel size comparison determines that the pixels falls between the two maps where the sizes most

-10-

closely correspond. The results of the comparison is a number consisting of an integer and a fraction, such as 3.2, which represents the apparent distance 16 of the object 13 to the viewer 10. Thereafter, for

5 trilinear interpolation, an interpolation of texture values that correspond to the four texels within each respective map that surround the projections 17a and 17b, respectively, is performed to produce a single texture value in each map. The interpolation within

10 each map could be any 2-D interpolation method, including bilinear interpolation. The two computed values are then utilized as input to a 1-D interpolation between the two results. It is observed that this method represents the most realistic method

15 of texturing polygons in a computer generation system. This method, while it is extremely accurate, it is computationally expensive for today's desire to design 3D rendering computer chips for use in PCS.

FIG. 2 is a flowchart describing the general

20 method steps which constitute the present invention. The method is started at step 20 by enabling texture mapping for pixels covered by an object. At step 22 a texture LOD value is computed for the pixel of interest and can be represented as consisting of an

25 integer component, intLOD, and a fractional component, fracLOD (i.e. $LOD = intLOD.fracLOD$). The integer component represents the nearest (from the viewer) LOD map to be used, $LOD=N$, while the fractional component represents the relative distance of the pixel LOD

-11-

value between the near and far texture maps. Prior to any potential contributions from the addition of a dithering offset, the near texture LOD map, $LOD=N$, would be selected to retrieve a texture value whenever
5 the fractional LOD component, $fracLOD$, is less than one half ($\frac{1}{2}$). When the fractional LOD is greater than or equal to one half the far texture LOD map, $LOD=N+1$, is selected to retrieve a texture value.

At step 24 a dithering offset is computed.
10 In the preferred embodiment, the dithering offset computed by first determining the display (I,J) coordinate of the pixel of interest and using the (I,J) address as an index into a table-look up to retrieve a dither offset value. Alternatively,
15 combinational logic could be used to calculate the dithering offset value based on the location of the pixel. The dithering offset computation could be based on the location of the pixel in a span of pixels if polygons are being rendered using span processing. At
20 step 26 the computed dithering offset value is applied to the computed LOD value from step 24 and the result is the dithered pixel LOD value. The dithered pixel LOD value will also be in the form of integer and fractional values. The LOD selection is then made in
25 step 28 by either rounding off where .5 or greater selects the higher LOD number and less than .5 selects the lower LOD. Truncation can also be used where the LOD is selected based solely on the integer value. It is observed applying the dithering offset has the

-12-

effect of "pulling" the computed texture LOD values for some pixels to the near integer texture map value, $LOD=N$, that otherwise would have been selected from the far map. Similarly, the offset may "push" other
5 computed LOD values to the far texture LOD map, $LOD=N+1$ that otherwise would have selected the near map. When round-off is used, pulling to the near texture map occurs when the fractional portion of the result of step 26 is less than one-half the distance
10 between the near and far texture maps. A push to the far map occurs when the fractional portion of the result of step 26 is equal to or exceeds one-half the distance between the two texture maps.

FIG. 3 illustrates by example the
15 probability of a computed pixel LOD value being assigned to either the near texture map, $LOD=N$, or the far texture map, $LOD=N+1$, for a selected dither range of .4. There are many dithering offset ranges that will have a dither range of .4. One such range is (-.2
20 to +.2) and another is (0 to .4). The graph of Fig. 3 shows the probability for the dithering offset range of (-.2 to +.2) and rounding is used. If truncation is used, the graph would still apply if a bias of .5 was added to the dithering offset. The application of bias
25 values will be explained in connection with Fig. 7. In addition, the graph would also apply for a dithering offset range of (0 to .4), if a bias is added. For rounding, the bias must be (-.2) and for truncation the bias must be (+.3).

-13-

In this example (offset range $-.2$ to $+.2$), the dithering offsets are added to the computed pixel LOD value. It is observed that all pixels with computed LODs, prior to the addition of a dither offset, of less than $N+.3$, will be automatically assigned to the near texture map, LOD N , with a probability of 100%. The dithering offset, given the dither range of $.4$, has no effect on these computed values. It is also observed that all pixels with computed LODs $N+.7$ and larger will be automatically assigned to the far texture map, LOD $N+1$. Again, the dithering offsets will have no effect on these computed pixel LOD values. Only those pixels whose computed LOD values lie in the range of $(N+.3$ to less than $N+.7)$ will have some probability of being assigned to a different LOD map because of the addition of a dithering offset. By example, pixels with a computed LOD of $N+.5$, have a 50% probability of being assigned to either LOD N or LOD $N+1$. It is observed widening the dither range would affect a larger percentage of computed pixel LOD values as candidates for selection to either map N or $N+1$. In the example shown only those computed LOD values in the narrow range of $N+.3$ to less than $N+.7$ are candidates to be pushed or pulled, all other values are immediately assigned to a particular map irrespective of the contribution of the dithering offset.

-14-

FIG. 4 illustrates the probability of a computed pixel LOD value being "pushed" or "pulled" for the dither range of 1. Again, the dither range can be achieved by many offset ranges, such as, (-.5 to +.5) or (0 to 1). The graph shows the probability for a range of dithering values of (-.5 to +.5) if rounding is used or if truncation is used this same range plus a bias of +.5. For a range of (0 to 1) the graph of Fig. 4 applies if truncation is used or if rounding is used a bias of (-.5) must be added. In this example, the computed LOD texture values for all computed pixel LOD values would be candidates to be either pulled into the near texture map or pushed into the far texture map. There is no equivalent region of automatic allocation as illustrated in FIG. 3 by the ranges. It is therefore apparent, from the examples illustrated in FIGs 3 and 4 that by adjusting the range of the dithering offsets the region of the computed LOD texture values can be controlled.

FIG. 5 illustrates transitions between LODs for the dither range of .4 for a rectangular polygon view in perspective. As is well known in the art, the apparent distance is a function of the viewer to polygon distance and the orientation of the polygon with respect to the view ray from the observer to the polygon. The pixels in the shaded transition areas of the polygon are dither blends of adjacent integer LODs. All of the pixels in the white areas of the polygon are assigned to an nearest integer LOD.

-15-

FIG. 6 is a graphical illustration of the contribution of a dithering offset to a computed pixel value for 3 representative ranges of dithering : 1) full dithering case [diagonal line] - dither range of 1 (-.5 to +.5), 2) no dithering [solid stair-step] - dither range of 0 (0-0), and an intermediate case, 3) partial dithering [dotted line] - dither range of .25 (-.125 to +.125). The first case, full dithering, is equivalent to what would result in a true trilinear interpolation mode, where each computed LOD value is a candidate for being pulled or pushed into the near or far texture map. As noted above, this may result in unwanted artifacts. The second case, solid staircase line, describes the effect of no dithering. In this case, computed pixel values abruptly transition or jump from one LOD map to the next at some predefined threshold value, typically one-half the distance between the two LOD maps which bracket the object. The third case, represented by the dotted line, represents a dither range of .25. In this case the addition of a dither offset will have no effect on 3/4 of the computed pixel LOD values. More particularly, the method would not effect those computed pixel offset values whose fractional component, prior to the addition of a dithering offset, is greater than or equal .75.

FIG. 7 is a block diagram illustrating the general components of a dithering unit apparatus 70 which selects a level of detail (LOD) texture map for

-16-

each pixel covered by the graphics primitive. The apparatus requires as input the coordinates in screen space of the pixel being texture. For example, the 2 least significant bits 71 of the I and J screen
5 coordinates of the pixel to be textured could be used to represent the location of the pixel. These two values are input as an index into the two-dimensional dither pattern generator 72. The generator 72 may take the form of a table look-up containing the pattern or
10 array of dithering offsets. Alternatively, the generator 72 could be comprised of combinational logic to generate the offsets for different textures or polygons. The function of the pattern generator is to return a single dither offset value based on the
15 address bits of a particular pixel in screen space.

For example, in one embodiment, a table look-up is used in a system that processes polygons a span at a time. The table look-up 72 is organized to mirror the span configuration, such as a 4x4 array,
20 thereby reflecting the physical dimensions of a span. The dithering offset retrieved by the table 72 is determined by utilizing the position of the pixel's position in the current span as an index to the table 72.

25 From the table look-up 72 a single dither offset value 73 is output and supplied as input to a dither value modulator 74. The modulator 74 is optional. The modulator 74 permits several dithering offset ranges to be available from only one set of

-17-

values stored in the look-up. The degree of modulation applied to the dither value 73 is determined by a second input to the dither value modulator 74 as a control state variable 77. The
5 modulator 74 can be programed to increase or decrease the stored offset values. For example, the modulator 74 can be an attenuator. In that case, the control state variable 77 can be set to apply dithering offsets in the range of zero to full attenuation, with
10 intermediate values of 1,2, and 3 correspond to one-half, one-fourth and one-eighth attenuation respectively. For example, the control state variable 77 can be 3 bits that are encoded which can provided 8 shifts but only 5 are used. The dithering offset value
15 73 is a 4 bit number. The attenuator will then shift this number to the right by the amount of attenuation, and 0 is inserted in the leading bits. The attenuation can be 0 shifts (full weight), 1 shift (half) 2 shifts (one-fourth 3 shifts (one-eighth) or 4 shifts
20 (completely shifted, therefore disabled).

The modulated dither value 75 is then input to an adder unit 76 along with the computed pixel LOD value 78 to provide the dithered pixel LOD value 82. Shown in Fig. 7 is an optional bias means. In this
25 embodiment, an LOD bias state variable 79 is also input to the adder 76. The bias state value is input to permit a truncation of the final added result instead of the more computationally expensive rounding operation which would have to be performed without it

-18-

to yield similar results. The bias state value can also be used to change the position of where the transitions from one LOD to another occur.

5 The adder 76 outputs a dithered pixel LOD value 82 with an integer and fractional component. The dithered pixel LOD value is then input to means 80 for performing the mathematical operations of either truncation or rounding to yield an integer number only. The integer portion 81 of the dithered pixel
10 LOD value is output for selecting an LOD map.

FIG. 8 is a representation of the internal configuration of the two dimensional texture dither pattern table. The dimensions of the look-up table 90 are matched to the dimensions of a span (4x4) so
15 that selection of a dither offset value from the look-up table for a particular pixel is made by indexing the two least significant bits of the pixel's I and J address in screen space into the table. In the illustrative embodiment, a single look-up table is
20 used for each processed span. However, alternate embodiments could create a dither offset dedicated to each and every pixel in the display space. The dithered offset values 92 stored in the table look up 90 are four bit values ranging from zero to 15/16. In
25 the illustrative embodiment these sixteen values are chosen to be evenly distributed in the selected dither range, however, the method may be implemented to reflect any user selected distribution of values with

-19-

certain distributions yielding more visually appealing results than others.

While several embodiments and variations of the present invention are described in detail herein,
5 it should be apparent that the disclosure and teachings of the present invention will suggest many alternative designs to those skilled in the art and that the scope of the invention should be limited only by the claims appended hereto.

10

-20-

CLAIMS:

Having thus described our invention, what we claim as new and desire to secure by Letters Patent is:

- 5 1. In a computer image generation system for generating an image of an object by controlling the value of pixels forming the image of the object, the value for each pixel being derived from stored data including texture data for applying texture to the
10 object, the texture data being stored in a sequence of texture maps consisting of a plurality of levels of detail (LODs) versions of the texture data, a method of transitioning between LODs for applying the texture to the object, the method comprising:
- 15 computing a texture LOD value for each pixel representing the image of the object;
 providing a dithering offset value for each pixel;
- 20 applying said dithering offset value to each computed LOD value for each pixel resulting in a dithered pixel LOD value for each said pixel; and
 outputting the dithered pixel LOD value for each pixel for selecting an associated texture LOD map.
- 25 2. The method of claim 1 wherein said step of providing a dithering offset value includes selecting a dithering offset value from said range of dithering offset values based on the location of the pixel.

-21-

3. The method of claim 2 wherein the step of selecting includes basing the selecting on the location of the pixel in a span of pixels.
- 5 4. The method of claim 1 further including the step of modulating the provided dithering offset value.
- 10 5. The method of claim 4 wherein the step of modulating includes attenuating the provided dithering offset value.
- 15 6. The method of claim 5 wherein the step of attenuating includes shifting bits representing the dithering offset value based on a control state variable.
- 20 7. The method of claim 4 wherein the step of attenuating is provided in discrete steps from full attenuation to zero attenuation.
- 25 8. The method of claim 1 the step of applying includes adding a bias value to the dithered pixel LOD value.
9. The method of claim 1 wherein the step of applying includes adding the dithering offset value to said computed pixel LOD value.

-22-

10. The method of claim 1 further including computing an integer value from said dithered pixel LOD value comprised of an integer and a fraction by one of truncating and rounding.

5

11. In a computer image generation system for generating an image of an object by controlling the value of pixels forming the image of the object, the value for each pixel being derived from stored data including texture data for applying texture to the object, the texture data being stored in a sequence of texture maps consisting of a plurality of levels of detail (LODs) versions of the texture data, a method of transitioning between LODs for applying the texture to the object, an apparatus for determining the LODs to be used for each pixel that reduces artifacts when transitioning between LODs when applying texture to the object, the apparatus comprising:

15

a dither pattern function generator for generating a dithering offset value for each pixel representing the image of the object to be textured; and

20

a combiner for receiving a computed pixel LOD value and a dithering offset value for each said pixel, said dithering offset value being received from said dither pattern function generator, and for combining said computed pixel LOD value and dithering offset value to produce a dithered pixel LOD value.

25

-23-

12. The apparatus of claim 11 wherein said dither pattern function generator includes a table look-up.
- 5 13. The apparatus of claim 11 wherein said dither pattern function generator includes means for selecting a dithering offset value from said table look-up based on the location of the pixel.
- 10 14. The apparatus of claim 13 wherein the number of dithering offset values in said table look-up being equal to the number of pixels in a span of pixels, and said means for selecting includes means for basing the selecting on the location of the pixel in said span of
15 pixels.
15. The apparatus of claim 11 further including a modulator coupled between said dither pattern function generator and said combiner.
- 20 16. The apparatus of claim 15 wherein modulator includes means for attenuating the dithering offset value.
- 25 17. The apparatus of claim 16 wherein the means for attenuating includes means for shifting bits representing the dithering offset value based on a control state variable.

-24-

18. The apparatus of claim 17 wherein the means for attenuating includes means for providing the attenuation in discrete steps from full attenuation to zero attenuation.

5

19. The apparatus of claim 11 wherein said combiner includes means for adding a bias value to the dithered pixel LOD value.

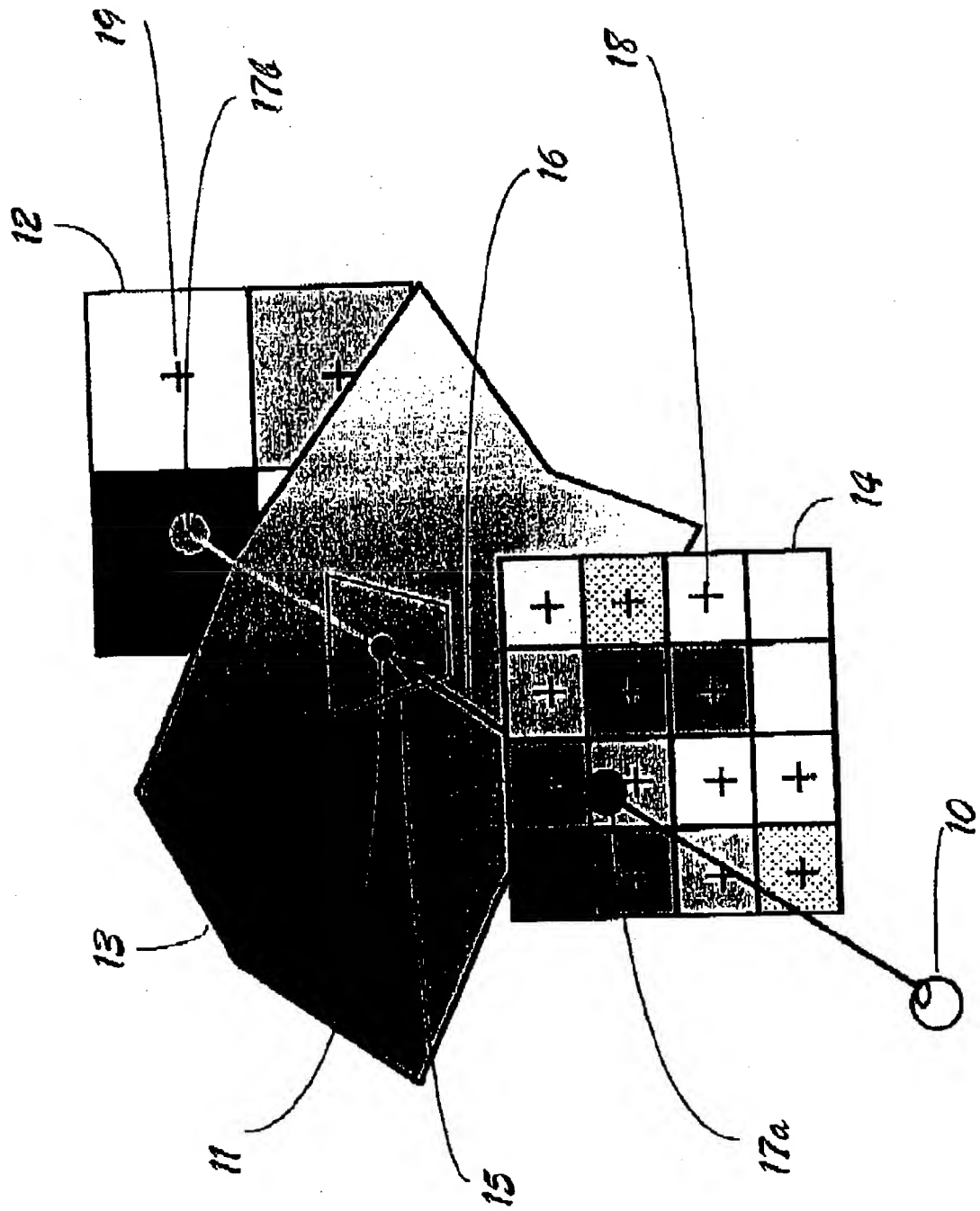
10

20. The apparatus of claim 11 wherein said combiner is an adder.

15

21. The apparatus of claim 11 further including means for computing an integer value from said dithered pixel LOD value comprised of an integer and a fraction by one of truncating and rounding.

Figure 1



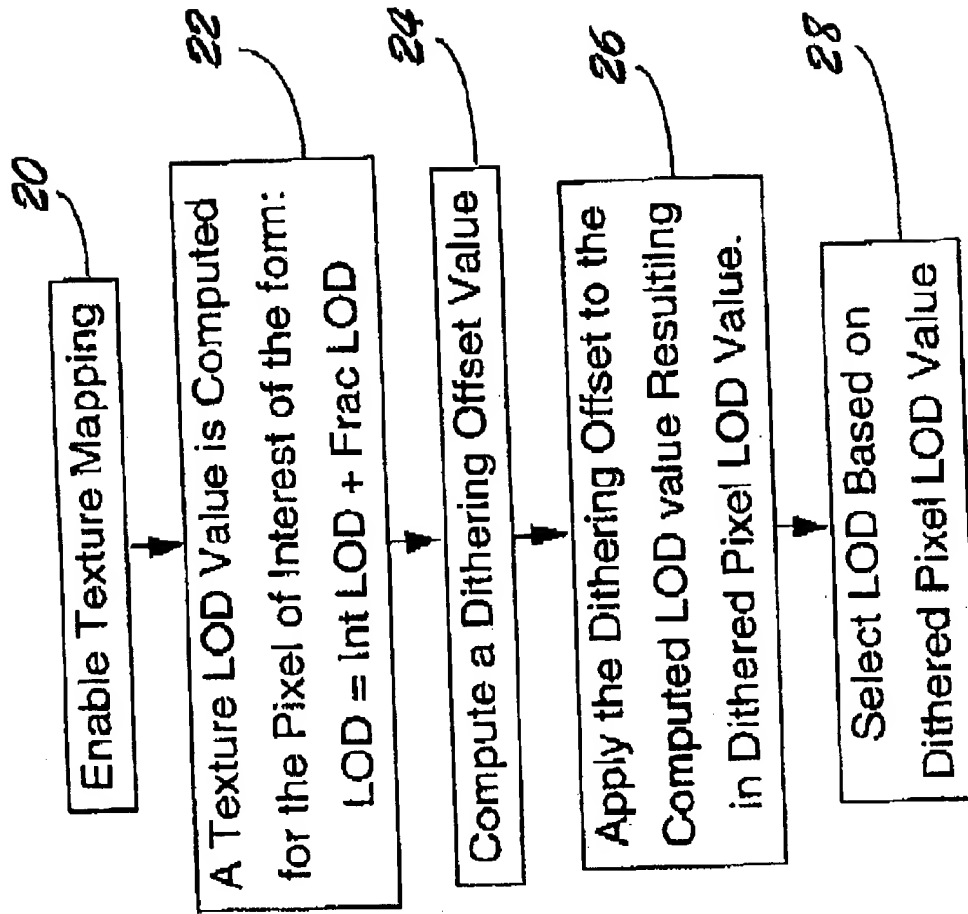
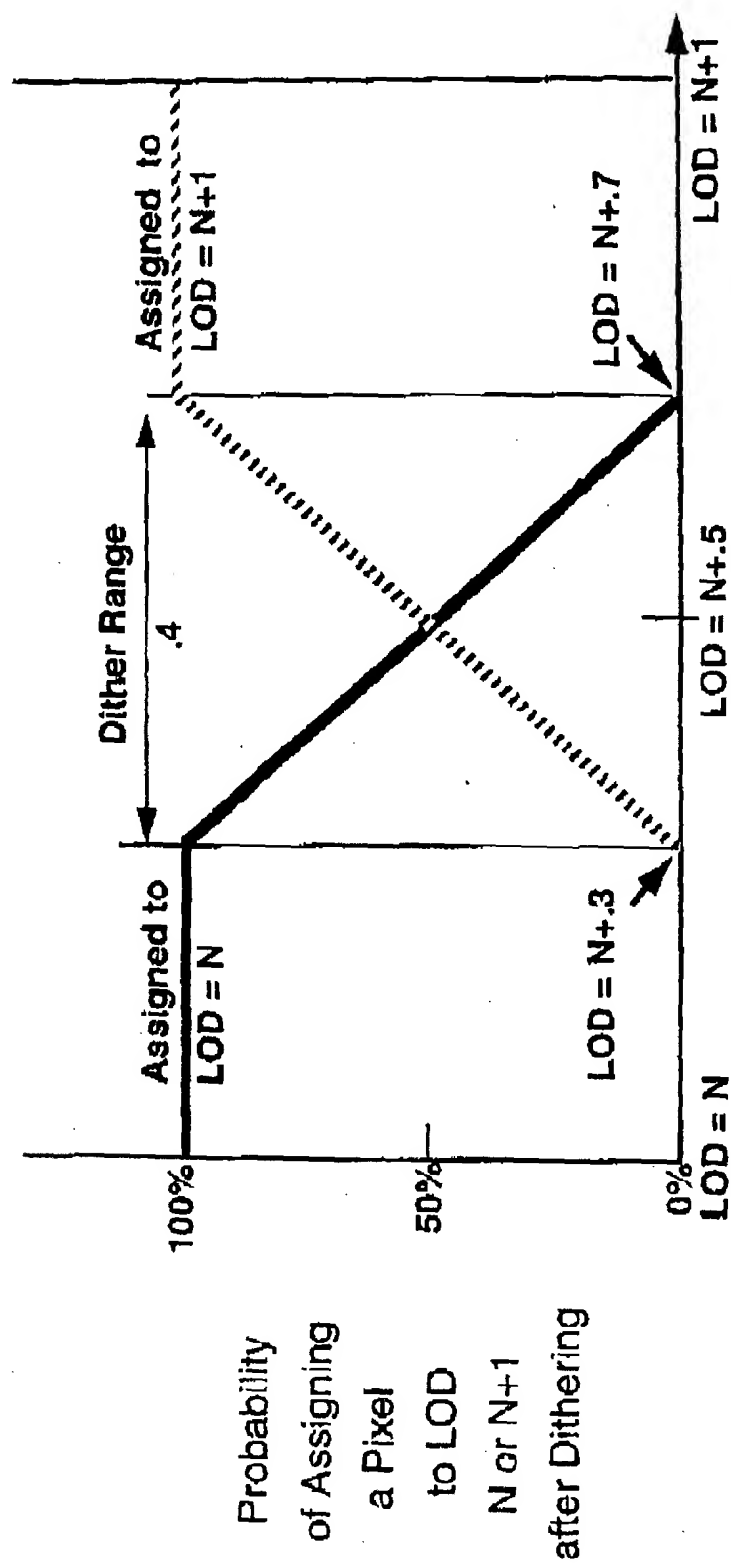
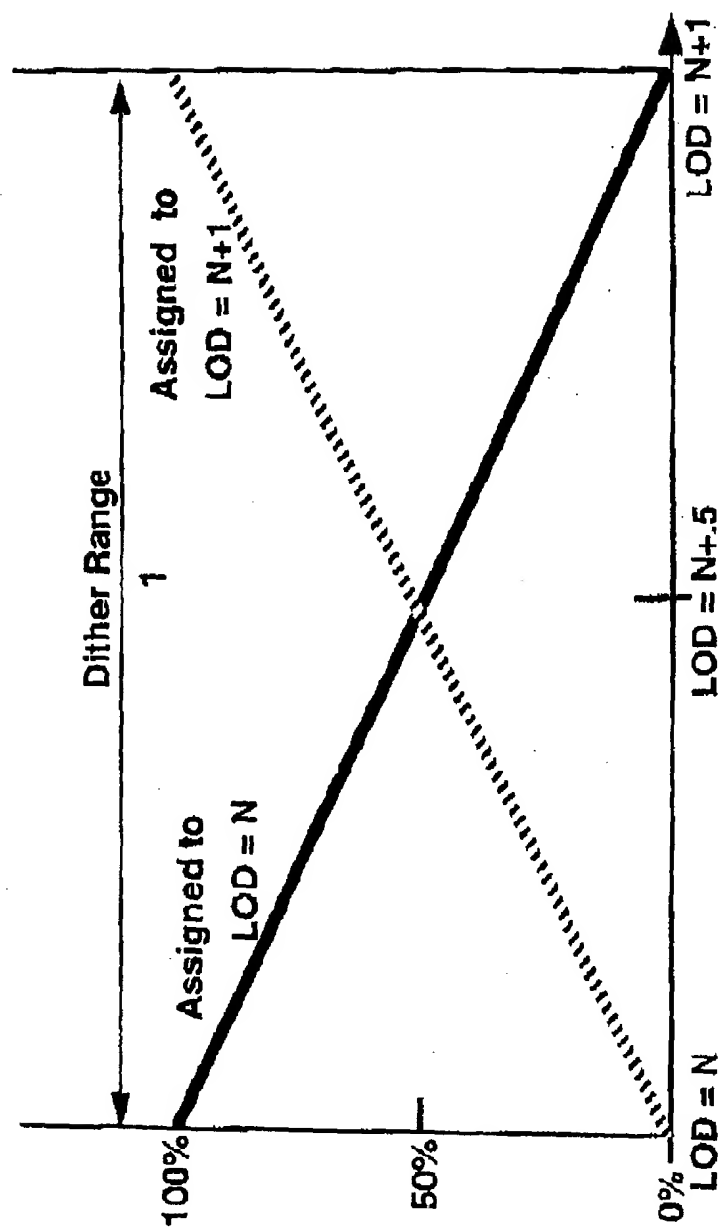
**Figure 2**

Figure 3



Computed Pixel LOD Value Based on Apparent Distance
(Log 2 of Texels/Projected Pixel)

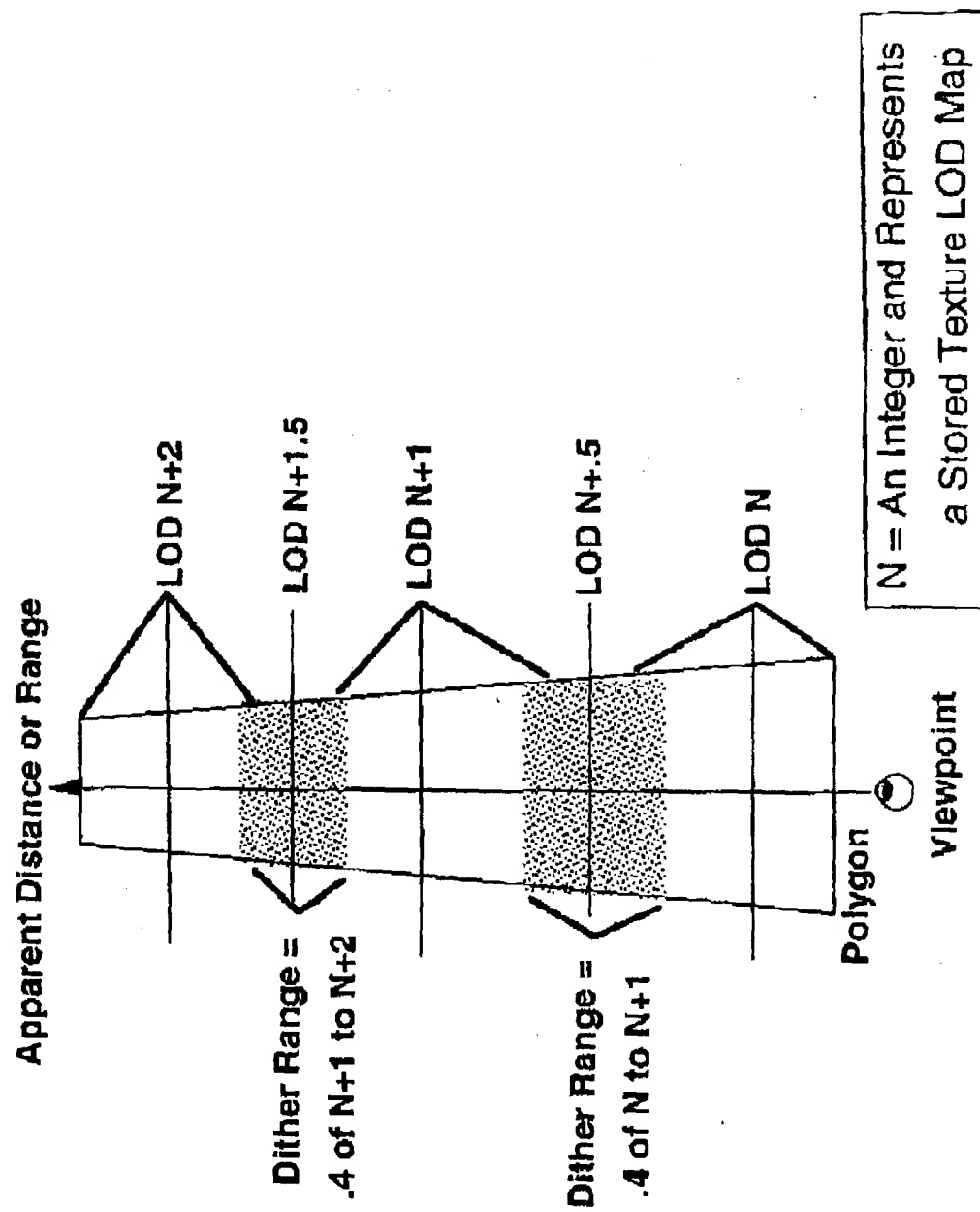


Probability
of Assigning
a Pixel
to LOD
N or N+1
after Dithering

Computed Pixel LOD Value Based on Apparent Distance
(Log 2 of Texels/Projected Pixel)

Figure 4

Figure 5



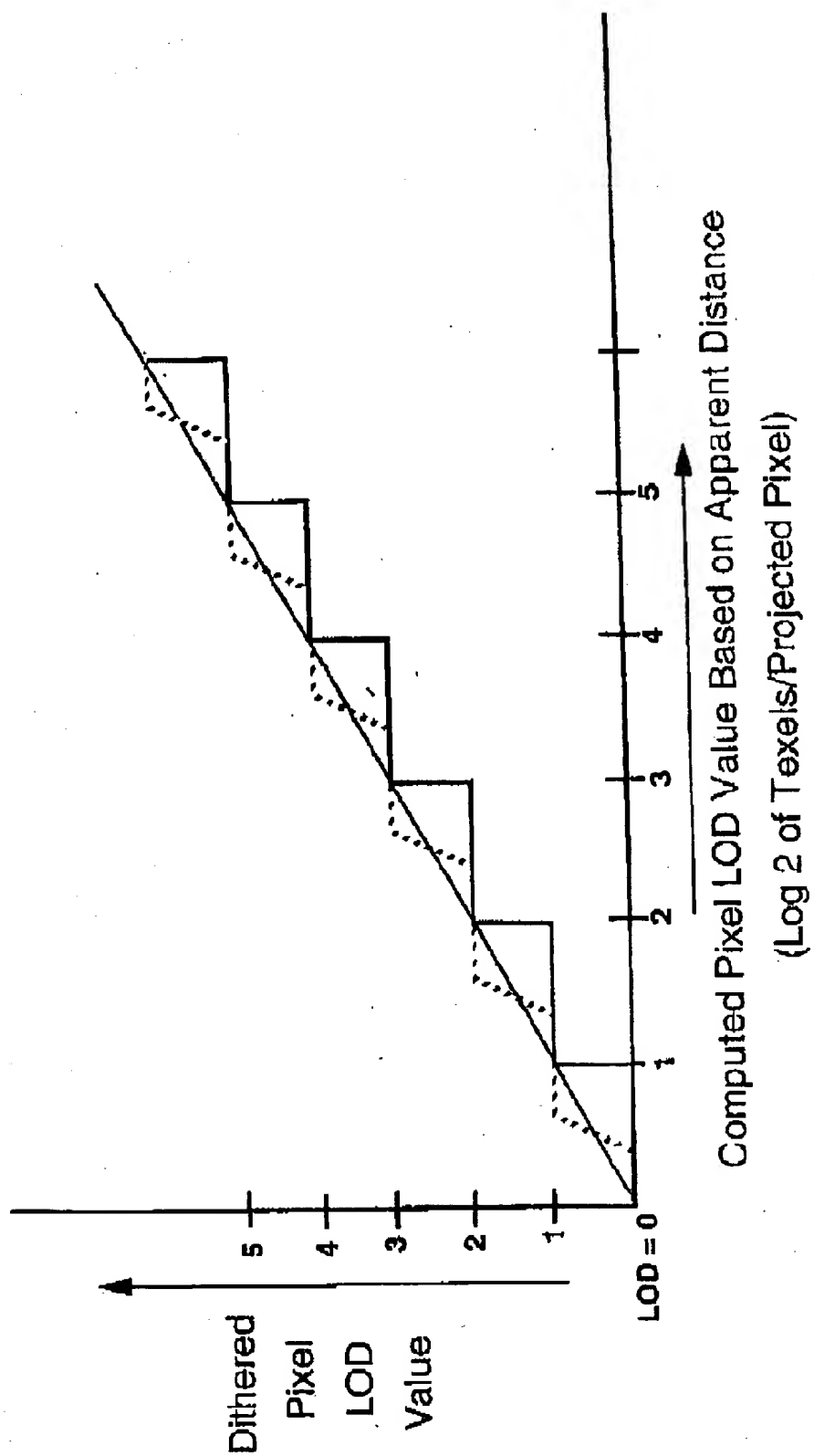
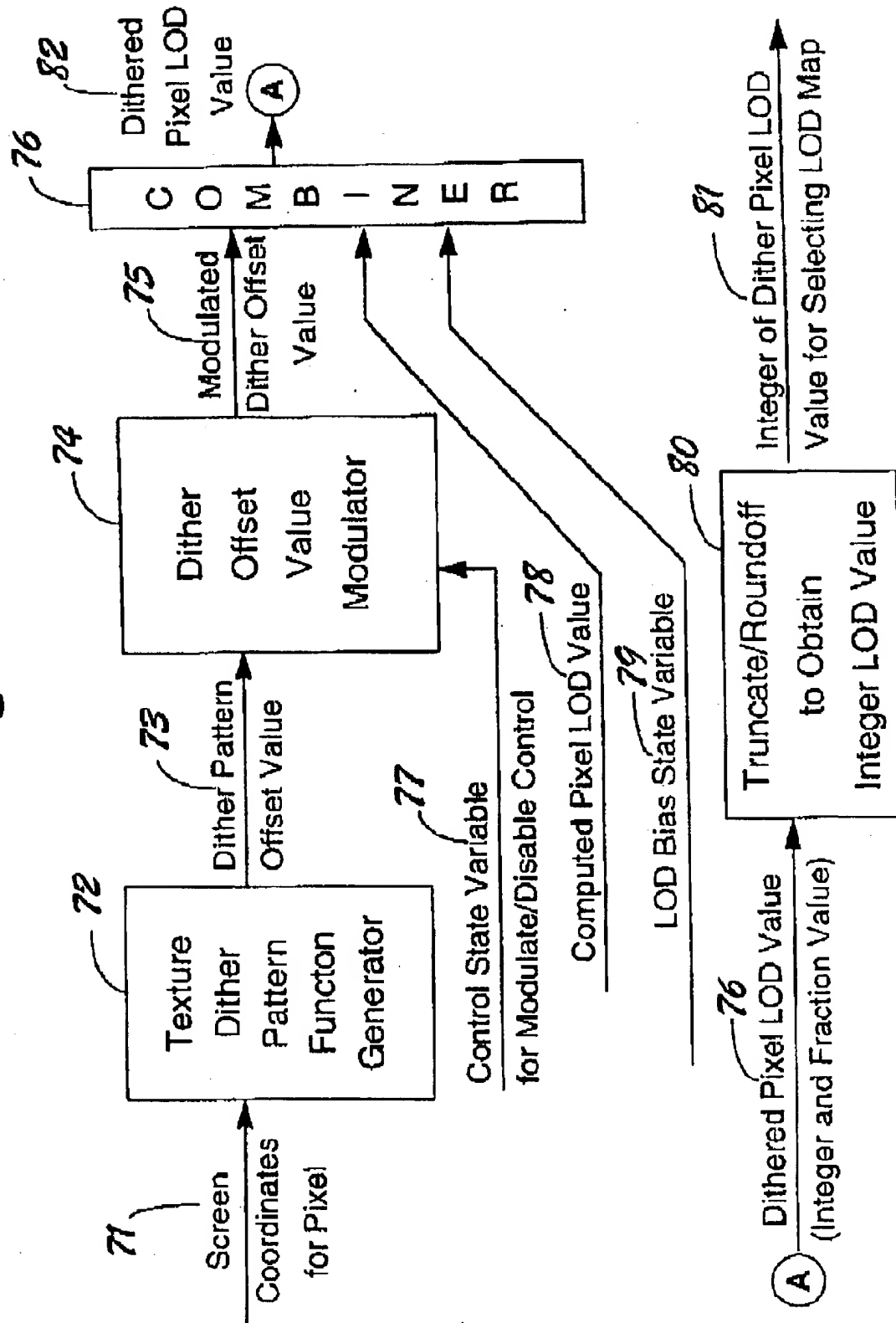
**Figure 6**

Figure 7



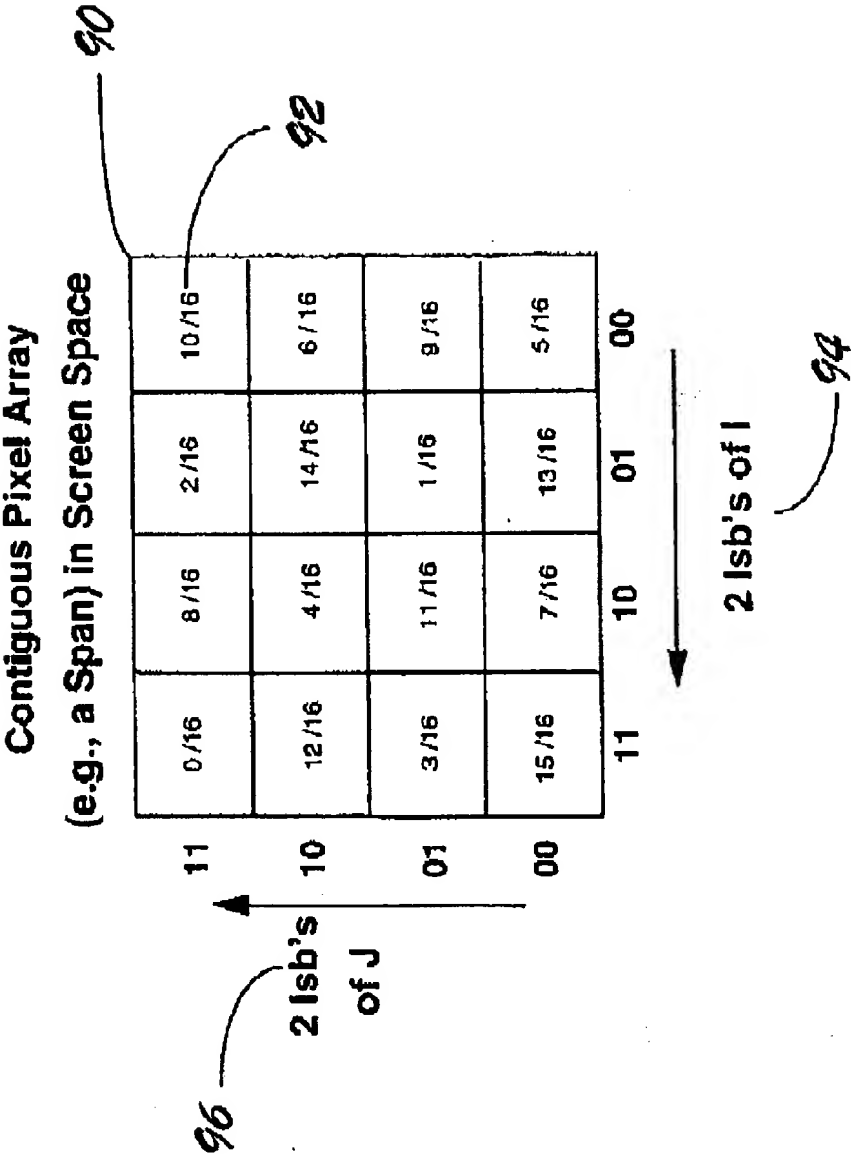


Figure 8

METHOD AND APPARATUS FOR
TEXTURE LEVEL OF DETAIL DITHERING

ABSTRACT

5

A computationally efficient method for
minimizing the visible effects of texture LOD
transitions across a polygon. The minimization is
accomplished by adding a dithering offset value to the
10 LOD value computed for each pixel covered by a
graphics primitive to produce a dithered pixel LOD
value. The dithering offsets may be generated from a
table look-up based on the location of the pixel
within a span of pixels. The dithered pixel LOD value
15 is used to as an index in the selection of a single
LOD texture map from which a textured pixel value is
retrieved. The range of dithering offset values can be
adjusted by modulating the values in the table look-
up.

20